Fusing BERT and BiLSTM Model to Extract the Weaponry Entity

Haojie Ge, Xindong You, Jialai Tian, and Xueqiang Lv(✉)

Beijing Key Laboratory of Internet Culture Digital Dissemination, Beijing Information Science and Technology University, Beijing, China
lxq@bistu.edu.cn

Abstract. Weaponry entity extraction is an indispensable link in the process of constructing a weaponry knowledge graph. In terms of entity extraction of weapons and equipment, a fusion model of domain BERT model and BILSTM model with embedded word vectors and word conversion rate vectors is proposed to identify weapons and equipment entities. First, the BERT model is used to perform pre-training tasks on massive weaponry corpus. Secondly, the Word2vec model is used to train the word vectors to provide a priori semantic information, and the word conversion rate vector is embedded to input more a priori information to the model. Finally, the hierarchical entity extractor extracts entities of different categories. Experiments results show that the fusion model has strong coding ability and sufficient prior knowledge, and the F1 value on the Global Military Network corpus reaches 91.436%.

Keywords: Weaponry entity extraction · BERT · BILSTM · Word conversion rate vector · Hierarchical entity extractor

1 Introduction

If the military builds a knowledge graph in the field of weapons and equipment, then they will manage its weapons and equipment more efficiently. In other words, this knowledge graph helps the military analyze its combat effectiveness, and obtain more information on the weapons and equipment of competitors. There are a large number of unstructured texts containing weapon and equipment information on the Internet, which can be extracted to form a weapon and equipment knowledge graph. Named entity recognition (NER) is an indispensable step in building a knowledge graph of weapons and equipment.

Xiang Xiaowen [1] used the Hidden Markov model to carry out the Chinese named entity recognition task. To fill the deficiencies of the model, its results were used with rules to improve the recognition effect. The rules are as follows: selection rule, split rule, supplementary call rule, boundary repair rule, consolidation rule. Feng Yuntian et al. [2] used a conditional random field model to extract named entities from military texts, and added feature templates and domain dictionaries to provide prior information to improve

Y. Weng et al. (Eds.): TridentCom 2020, LNICST 380, pp. 95–107, 2021.
https://doi.org/10.1007/978-3-030-77428-8_8

the extraction effect of military entities. In 2019, Li Jianlong [3] and others analyzed the difficulties of military text entity recognition and used a Bidirectional LSTM recurrent neural network model to solve the recognition problem of named entities in the military field. By adding a combination of word vectors and attention mechanisms. The LSTM recurrent neural network model is improved the recognition effect Ju Jiupeng et al. [4] used the method of combining conditional random fields and rules to identify the geographical space named entities in the text, using the BIO tagging scheme, adding part of speech, linguistic features, and text features to the feature template of the conditional random field, and adding hierarchical rules, company name rules, place name rules to improve the extraction effect. Zhang Hainan [5] used a deep neural network to perform named entity recognition tasks on the People's Daily corpus. To solve the problem of word segmentation tool segmentation errors and sentence length differences after segmentation affecting the network effect, the word hybrid embedded matrix was added to improve the extraction effect.

With the development of neural networks, the LSTM neural network model is widely used. Its ability to remember or forget text information determines its competence for natural language processing tasks. More and more scholars begin to use the LSTM neural network model to solve the problem of named entity recognition. Mourad Gridach [6] used the combination of LSTM neural network and conditional random field to recognize biological named entities, and the precision value is up to 90.27%. Ji Xiangbing [7] and others used the Attention-BILSTM method. The Attention mechanism can not only solve the long-distance dependence problem of traditional neural networks, but also increase the model's attention to special words, and give greater attention to important words to improve the effectiveness of the model. The language model has advanced the NLP tasks greatly. Google proposed the BERT [8] pre-training language model in 2018. BERT can be fine-tuned to adapt to various NLP tasks. Google has proved the effectiveness of BERT in various NLP tasks. BERT greatly improves the task of named entity recognition. Wang Ziniu [9] used BERT for the task of named entity recognition. It input the output vector of BERT into the LSTM-CRF layer to further extract the features and achieved an F1 value of 94.86% on its corpus.

The corpus of this paper is based on military texts. Military texts have their domain specificities. Because the texts are from the Internet, some irregular or networked names will increase the difficulty of recognition. At present, named entity recognition in the general field has received extensive attention and widely research, and has had practical effects, but few scholars researched named entity recognition in the military field. This paper collects a large number of military texts on the Internet, uses the BERT language model for pre-training based on this, and merges it with the BILSTM model with embedded word vectors and word conversion rate vectors, and finally uses a hierarchical entity extractor to extract entities to solve military text naming entity recognition problem.

2 Design of Weapon Equipment Entity Extraction Model

Figure 1 is a weapon and equipment entity recognition framework based on the fusion of BERT and BILSTM.

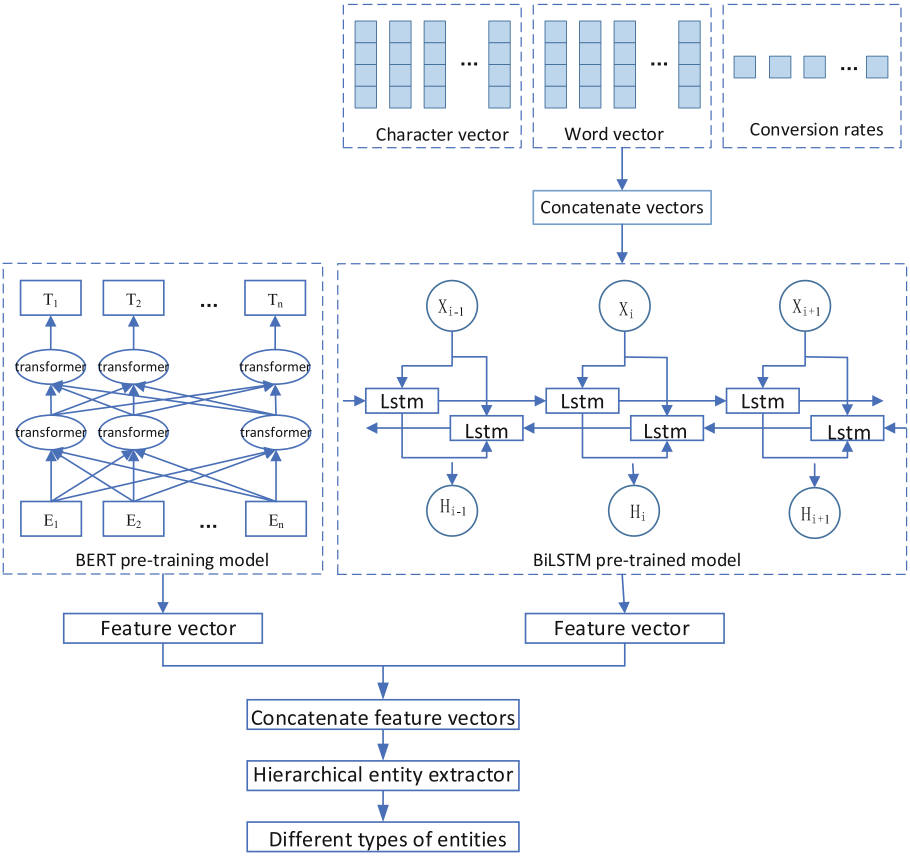

Fig. 1. Weapon and equipment entity recognition framework based on the fusion of BERT and BILSTM

This framework is based on the fusion of the BERT model and the BILSTM model to solve the problem of named entity recognition in weapon equipment. First, we pre-train the BERT language model on large-scale military texts and generate feature vectors after encoding the text through the pre-training BERT language model. Second, we use the Word2vec language model to train the word vector matrix, calculate the word conversion rate in the text to form a word conversion rate vector matrix, convert the text into a vector through the word vector matrix and the word conversion rate vector matrix, and input the vector into the BILSTM model to obtain the feature vector. Finally, the feature vectors output by the BILSTM model and the BERT language model are added to obtain the total feature vector, and the hierarchical entity extractor is used to extract the total feature vector to obtain different types of entities.

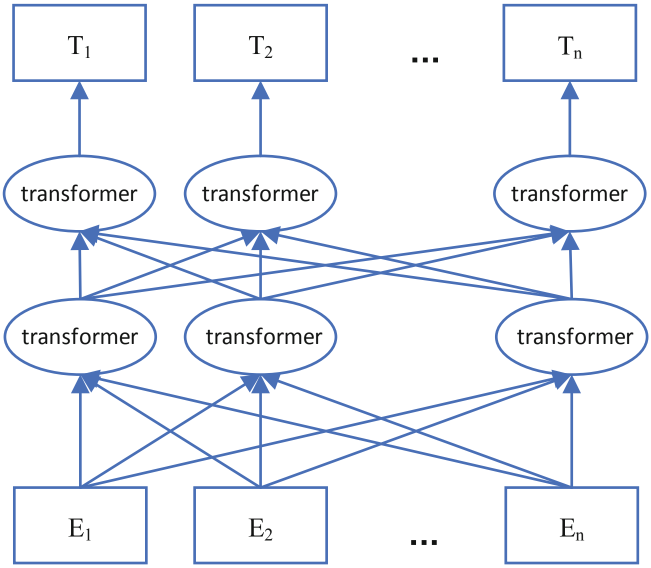

Fig. 2. Structure diagram of BERT model

2.1 BERT Pre-trained Language Model

The method of the BERT model adopts self-supervised learning which refers to supervised learning on the corpus without manual annotation. Using the BERT model to do pre-training tasks on a large number of military texts can encode the texts into vectors with military semantics, increase prior information and improve the model's ability to recognize weapons and equipment entities. Google has proved that the BERT model can transfer learning when the pre-training is completed, and it only needs to fine-tune the model input and output and model parameters when performing other NLP tasks.

The internal structure of BERT is shown in Fig. 2. The sentence gets E_1 to E_N vectors through word vector embedding and position embedding, and the final feature vectors T_1 to T_N are obtained through the transformer layer. The transformer includes an encoding structure and a decoding structure. It was originally used for machine translation tasks. However, it is applied to many NLP models at present because of its powerful feature extraction capabilities. In BERT, only the encoding structure of the transformer is used, but its decoding structure is discarded in that when the encoding structure is implemented, the current word can view the information of the front and rear words, while the current word of the decoding structure can only view the information of the previous word. Obviously, in the entity extraction task requires a coding structure that can obtain bidirectional information for which the BERT model is adapted. The reason for the Transformer's strong feature extraction ability is its internal Multi-Head Attention mechanism. Multi-Head Attention contains Self Attention mechanisms which are shown in Fig. 3. The specific operations of the Self Attention mechanism are as follows: the first step is the H vector passes through three different fully connected layers to obtain three vectors Q, K, and V. Then the second step is to obtain the vector Q

$\times$ K_T after matrix multiplication of Q and K_T, which represents the degree of correlation between words and other words.

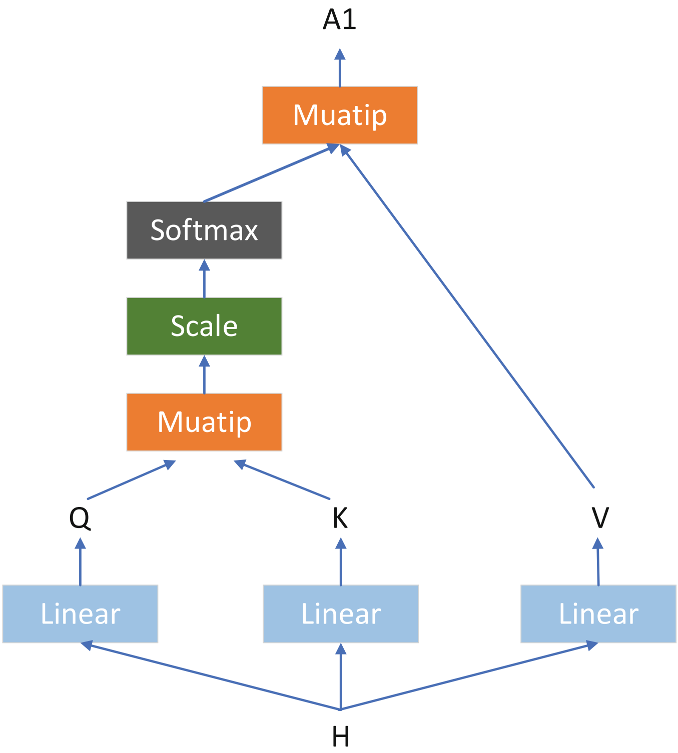

Fig. 3. Structure diagram of Self-Attention mechanism

The last step is to obtain the correlation degree vector between words by putting standardized Q $\times$ KT into the Softmax activation function, and then the correlation degree vector should be multiplied by V to obtain the vector A1. That is to say, each word in the BERT model will take into account the semantic weight of other words in the sentence when encoding, so it has a strong coding ability.

The pre-training of the BERT model includes two tasks. One is the task of predicting randomly masked words. Assuming that E_1 to E_N are known prior sentences, mask 15% of the words and let the model predict the masked words. The other task is to predict whether the next sentence belongs to the same article task as the current sentence. Supposed E_1 to E_{10} are the current sentence, and E_{10} to E_N are the next sentence, then the model try to predict whether E_1 to E_{10} and E_{10} to E_N belong to the same article. Experiments have proved that using the BERT model for pre-training tasks on military corpus can encode sentences into vectors that are closer to their real semantic information, thereby improving the effect of entity extraction.

2.2 BILSTM Input Vector Representation

At present, the neural network-based named entity recognition method misses many prior features in the text, such as word features. If we use word segmentation tools or domain vocabulary to segment the text to obtain phrases, then the length of the phrase after the segmentation will be smaller than the length after the word segmentation.

To ensure that the dimension of the vector obtained after the word embedding of the sentence is consistent, the word vector is copied to form a new copy of the word length. The word vector matrix and word vector matrix are trained by the Word2vec model. The shape of the word vector matrix is (N, 128), and N represents the number of words or the number of words, 128 is the predetermined vector dimension. After tagging the text, each word has a chance to be transformed into any label, and the transition probability of each word into each label can be calculated. The greater the transition probability, the greater the expected value of the corresponding label. By calculating the conversion probability of each word corresponding to the label, the conversion rate vector matrix is obtained. The calculation formula of transition probability is as follows:

$$p_{\text{label}}^{\text{word}} = \frac{Count_{\text{label}}^{\text{word}}}{Count_{\text{word}}} \tag{1}$$

$p_{\text{label}}^{\text{word}}$ is the probability that a word is transformed into a specific label, which is equal to the total number of times the word is converted to the label divided by the total number of times the word appears. For example, in military texts, the sentence is: the aircraft carrier Liaoning is my country's first aircraft carrier, the entity in the sentence is the aircraft carrier Liaoning, and the name of the aircraft carrier category entity generally ends with the word ship. Assuming that the conversion rate of $p_1^{舰}$ship is calculated to be 0.98, the probability that the word ship is converted into a 1 tag is 0.98, which means that the probability of the word ship as the tail text of the named entity is 0.98. The number of rows of the conversion rate vector matrix is the number of words N, the number of columns is the number of tags NUMtag, and the matrix shape is (N, NUMtag). The text is transformed into the input vector of the BILSTM model in the shape of (Lenseq, 256 + NUMtag) through the word vector matrix and the word conversion rate vector matrix. Lenseq represents the length of the sentence.

2.3 Fusion of BILSTM and BERT Models

The BILSTM model can capture the bidirectional information flow of the text, and it has been proved to be suitable for named entity recognition tasks. In this paper, the text vector is input into the BILSTM model to obtain the feature vector. The BILSTM model structure is shown in Fig. 4, from Xi − 1 to Xi + 1 are inputs vector. The input vectors are input into the model from the forward direction and the reverse direction respectively to obtain the feature vectors from Hi − 1 to Hi + 1.

The learning results of different neural network models have certain differences. For example, when the BERT model judges that the label of the ship character is 1, the probability is 0.98, while the BILSTM model judges that the probability of the ship character's label is 1 is 0.45. The label probability should be half of the sum of the

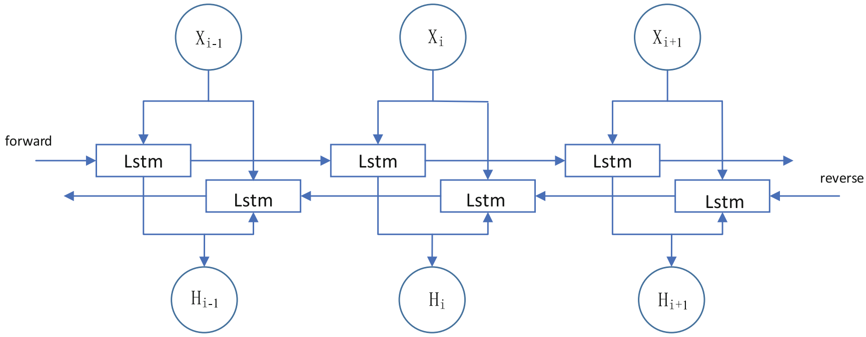

Fig. 4. Structure diagram of BILSTM model

probability results of the two models (0.715), and the predicted label of the ship character after the fusion of the two models should be 1, so the BERT model can correct the wrong results of the BILSTM model, and the BILSTM model can also correct the BERT model result. The feature vector of the BERT and the feature vector of BILSTM are added to obtain the feature vector with deeper semantic information of the text, which helps the hierarchical entity extractor to accurately extract the named entity.

Hierarchical Entity Extractor

When traditional neural networks use only one fully connected layer for sequence labeling, so only one label sequence can be generated per sentence. There are often multiple types of named entities in a sentence, and different types of named entities may overlap entities. For example, when the sentence is: The aircraft carrier Liaoning is China's first aircraft carrier, the target entity that needs to be extracted is the aircraft carrier Liaoning and Liaoning place named entities. Multiple fully connected layers are used to generate different types of labeling sequences to solve the overlapping problem of different types of named entities, which can also enhance the recognition ability of certain types of entities.

There are two types of labels in this paper: 0 and 1. A sentence will generate two-line tag sequences, one sequence is responsible for extracting the beginning text of the named entity, and the other sequence is responsible for extracting the ending text of the named entity. The sigmoid activation function is used to convert the output corresponding to each text into a value from 0 to 1. The size of the value indicates the degree to which the text is converted into a start text or an end text. When the corresponding value of the text is greater than 0.5, the text is taken as the start text or the end text.

The tag sequence generated by a sentence is shown in Fig. 5. The text extracted in sequence 1 is the beginning text of the entity, and the text extracted in sequence 2 is the ending text of the entity. The values 0.98 corresponding to the text in sequence 1 is greater than 0.5, so the text is extracted Liao is the beginning text of the entity, and the value 0.999 corresponding to the text ship in Sequence 2 is greater than 0.5, so the text ship is taken as the ending text of the entity. The final named entity extracted is the aircraft carrier Liaoning.

Fig. 5. Label sequence

3 Experiments

3.1 Dataset

The experimental data comes from the text data on ships and aircraft on the Global Military Network website, and finally, 5,000 sentences are marked. Among them, 4000 pieces are used as training data, 500 pieces are used as test data, and 500 pieces are used as verification data. The named entities of weapons and equipment in this article mainly consists of the following six categories: ships (the Nimitz aircraft carrier), aircraft (Black Hawk fighter), radar (Patriot radar), missiles (Sea Sparrow missile), systems (Aegis system), naval gun (11 m naval gun). An example of data labeled is shown in Fig. 6. The number 1 indicates that it is the beginning text of the weapon entity, and the letter after the "–" character is the English abbreviation of the entity category. For example, "jc" represents the ship category. The number 2 indicates that it is the ending text of the weapon entity, and "dd" indicates that the entity category is the missile category. The corresponding English abbreviations of entity categories are shown in Table 1, and the number of weapon and equipment entity categories in the corpus is shown in Table 2, where each weapon entity category accounts for an even proportion.

辽/1-jc	宁/0	号/0	航/0	空/0	母/0	舰/0	是/0	我/0	国/0
第/0	一/0	艘/0	航/0	空/0	母/0	舰/0	，/0	其/0	上/0
搭/0	载/0	爱/1-dd	国/0	者/0	导/0	弹/2-dd	节/0	省/0	燃/0

Fig. 6. Examples of the data tags

3.2 Evaluation Index

To verify the effectiveness of the model, the F1 value after the combination of precision P and recall rate R is used to judge the pros and cons of the model. The calculation

Table 1. English abbreviation for the different categories

Entity category	Entity category abbreviation
Ship	jc
Aircraft	fj
Missile	dd
Naval gun	jp
Radar	ld
System	xt

Table 2. Corresponding quantity of weapon entity category

Entity category	Quantity
Ship	3645
Aircraft	3412
Missile	2800
Naval gun	2588
Radar	2136
System	2689

formulas are shown in Eqs. (2), (3), and (4). When the names and categories of the predicted entity and the real entity are the same, it is called the total number of correctly identified entities.

$$P = \frac{\text{Total number of correctly identified entities}}{\text{Total number of entities identified}} \times 100\% \tag{2}$$

$$R = \frac{\text{Total number of correctly identified entities}}{\text{Total number of entities in the test data}} \times 100\% \tag{3}$$

$$F1 = \frac{2 \times P \times R}{P + R} \times 100\% \tag{4}$$

3.3 Experiments and Results Analysis

The model runs on the Ubuntu 16.04 operating system of the Dell server with 64G of memory. The GPU is 8 T V100 graphics cards, each of which has 16G of graphics memory, and the coding language is python3.6. We use the Keras deep learning framework. Through experiments, it is found that different model parameters correspond to different entity recognition results. The optimal parameters are finally determined by tuning as shown in Table 3.

Table 3. Parameter settings of the model

Parameter name	Parameter value
BERT word vector dimension	768
BERT pre-training Batch-Size	16
Maximum text length	256
BERT hidden layer dimension	768
BILSTM model layers	2
BERT-Dropout value	0.9
Number of BERT pre-training iterations	500
BILSTM-Dropout value	0.9
BILSTM word vector dimensions	200

To verify the feature extraction ability and transfer learning ability of BERT, we set up experiment 1 and 2. To prove that adding a layered entity extractor can more accurately identify weapon entities, we set up experiment 2 and 3. In all experiments, except for experiment 2, the layered entity extractor is not used, and other experimental models all use layered entity extractor. To verify that the BERT model pre-trained on the corpus of a specific domain is more suitable for named entity tasks than the pre-trained BERT model on the general domain corpus, we set up experiments 4 and 5. To verify the effectiveness of the BILSTM model combining embedded word vectors and conversion rate vectors, we set up experiment 3 and 5. The specific experiment content is as follows:

Experiment 1: uses the BILSTM model with embedded word vectors and word conversion rate vectors to perform named entity recognition tasks on the data set, and the name of the experiment is denoted as BILSTM-T.

Experiment 2: uses the BERT model that is pre-trained on the military text and has no hierarchical entity extractor to perform named entity recognition tasks on the data set. The experiment name is recorded as PRE-BERT-no hierarchical.

Experiment 3: uses the BERT model pre-trained on military texts to perform named entity recognition tasks on the data set, and the experiment name is recorded as PRE-BERT.

Experiment 4: uses Google's open-source Chinese pre-trained BERT model and the BILSTM model with embedded word vectors and word conversion rate vectors to fuse them on the data set to perform named entity recognition tasks. The experiment name is recorded as BERT-BILSTM-T.

Experiment 5: The BERT model pre-trained on military texts and the BILSTM model with embedded word vectors and word conversion rate vectors were used to fuse them on the data set for named entity recognition tasks. The name of the experiment is recorded as PRE-BERT-BILSTM-T.

The overall results of the five groups of experiments are shown in Fig. 7 and Table 4.

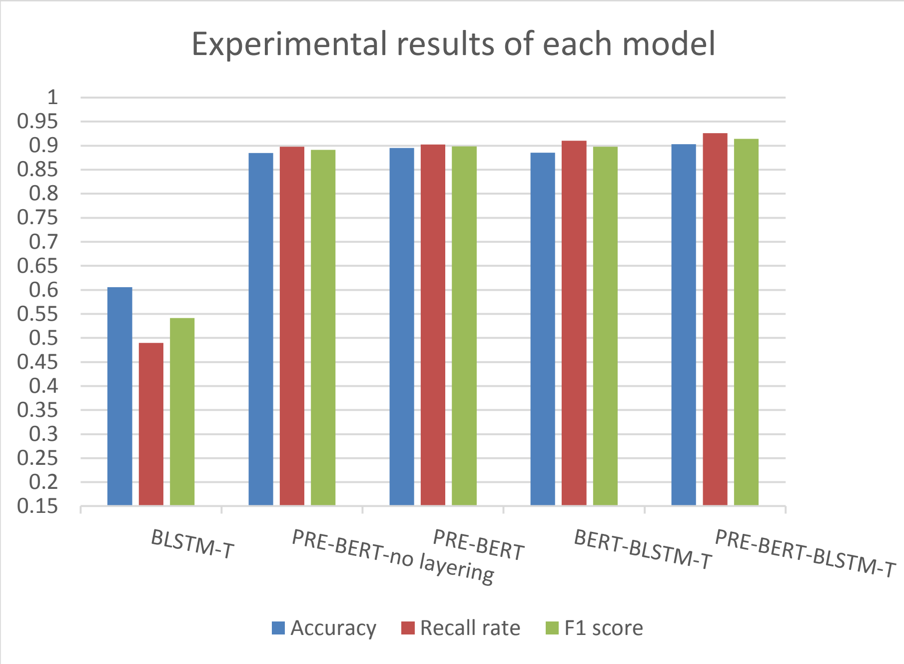

Fig. 7. Experimental results of each model

Table 4. Overall experimental results of the different models

Experiment no	Model name	Accuracy	Recall	F1 score
1	BILSTM-T	60.56	48.94	54.13
2	PRE-BERT-no layering	88.48	89.76	89.115
3	PRE-BERT	89.53	90.24	89.884
4	BERT-BILSTM-T	88.53	91.05	89.772
5	PRE-BERT-BILSTM-T	90.32	92.58	91.436

To verify the powerful feature extraction and transfer learning capabilities of the BERT model, a comparison between experiment 1 (BILSTM-T) and experiment 2 (PRE-BERT-no layering) is carried out. The experimental results show that the F1 value of the BILSTM-T model that uses only the embedded word vector and the word conversion rate vector is increased by 35 percentage points using the BERT pre-training model, indicating that BERT has strong transfer learning capabilities and feature extraction capabilities. Only with self-supervised learning on the unlabeled corpus of the domain and fine-tuning the model for supervised learning on a small amount of labeled text, the model can achieve good results.

To prove that the addition of a layered entity extractor can more accurately identify weapon entities, this paper conducted a comparative experiment between Experiment 2 (PRE-BERT-no layering) and Experiment 3 (PRE-BERT). Experiment 3 (PRE-BERT) a layered entity extractor is added based on experiment 2 (PRE-BERT-no layering). The accuracy and recall rate of the model with a layered entity extractor is slightly higher than that of the model without a layered entity extractor, indicating that adding a layered entity extractor can extract a certain type of entity more accurately.

To verify the effectiveness of the BILSTM model fused with embedded word vectors and conversion rate vectors, this paper conducted a comparative experiment of Experiment 3 (PRE-BERT) and Experiment 5 (PRE-BERT-BILSTM-T), Experiment 5 (PRE-BERT)-BILSTM-T) based on Experiment 3 (PRE-BERT), a BILSTM model that combines embedded word vectors and conversion rate vectors. Experiment 5 (PRE-BERT-BILSTM-T) compared with experiment 3 (PRE-BERT), the accuracy rate and recall rate increased by 0.79% and 2.34%, respectively. The recall rate increased significantly which shows that adding embedded word vectors and conversion rate vectors not only allows the model to acquire more prior features, but model fusion can correct the recognition errors of a single model, thereby improving the effectiveness of named entity recognition.

To verify the effectiveness of the pre-trained BERT model for specific domain corpus, this paper conducts a comparative experiment of experiment 4 (BERT-BILSTM-T) and experiment 5 (PRE-BERT-BILSTM-T) which based on Experiment 4 (BERT-BILSTM-T): the general domain corpus is changed to the military corpus. The accuracy and recall rate of the BERT model pre-trained with military corpus are both about 2% higher than that of the BERT pre-training model in the general field. It indicates that the pre-training BERT model on the military corpus can learn the prior features of the military corpus and thus increase the effect of named entity recognition in the military field.

Therefore, the BERT model pre-trained on the military text and the BILSTM model (PRE-BERT-BILSTM-T) with embedded word vectors and word conversion rate vectors are more suitable for the task of weapon equipment named entity recognition.

4 Conclusion

BERT model is used to conduct self-supervised training on the military corpus to obtain the military corpus pre-training BERT model and integrates the BILSTM model embedded with word vectors and word conversion rate vectors to solve the problem of weapon equipment named entity recognition. Experimental results show that the BERT model has strong transfer learning capabilities and feature extraction capabilities. The pre-trained BERT model on the specific domain corpus can encode sentences into vectors that are closer to their real semantic information, thereby increasing the effect of a domain named entities. Adding a layered entity extractor can extract a certain type of entity more accurately. The embedded word vector and word conversion rate vector can increase the prior features of the domain, and the model fusion makes the models correct each other. The model in this paper is effective for the task of weapon equipment named entity recognition.

Acknowledgments. This work is supported by National Natural Science Foundation of China under Grants No. 61671070. Project of Developing University Intension for Improving the Level of Scientific Research–No. 2019KYNH226, Qin Xin Talents Cultivation Program, Beijing Information Science & Technology University No. QXTCP B201908.

References

1. Xiang, X., Shi, X., Zeng, H.: A Chinese named entity recognition system combining statistics and rules. Comput. Appl. **25**(10), 2404–2406 (2005)
2. Feng, Y., Zhang, H., Hao, W.: Named entity recognition for military texts. Comput. Sci. **42**(7), 15–18, 47 (2015)
3. Li, J., Wang, P., Han, Q.: Military named entity recognition based on two-way LSTM. Comput. Eng. Sci. **4**, 20 (2019)
4. Ju, J., Zhang, M., Ning, J., et al.: Geospatial named entity recognition combining CRF and rules. Comput. Eng. **37**(7), 210–212 (2011)
5. Zhang, H., Wu, D., Liu, Y., et al.: Chinese named entity recognition based on deep neural network. Chinese J. Inform. **31**(4), 28–35 (2017)
6. LNCS Homepage. http://www.springer.com/lncs. Accessed on 21 Nov 2016
7. Ji, X., Zhu, Y., Li, F., et al.: Chinese named entity recognition based on Attention-BiLSTM. J. Hunan Univ. Technol. **5**, 14 (2019)
8. Devlin, J., Chang, M.W., Lee, K., et al: BERT: Pre-training of deep bidirectional transformers for language understanding (2018). arXiv:1810.04805
9. Wang, Z., Jiang, M., Gao, J., et al.: Chinese named entity recognition method based on BERT. Comput. Sci. **46**(11A), 138–142 (2019)